



TITLE:

多並列計算機向きかつ同期回数を半減させた前処理つき反復法の提案 (次世代計算科学の基盤技術とその展開)

AUTHOR(S):

村上, 啓一; 藤野, 清次

CITATION:

村上, 啓一 ...[et al]. 多並列計算機向きかつ同期回数を半減させた前処理つき反復法の提案 (次世代計算科学の基盤技術とその展開). 数理解析研究所講究録 2013, 1848: 25-32

ISSUE DATE:

2013-08

URL:

<http://hdl.handle.net/2433/195091>

RIGHT:

多並列計算機向きかつ同期回数を半減させた
前処理つき反復法の提案

A proposal of preconditioned iterative method with reduction
of synchronization suited to distributed parallel computers

村上啓一 (九州大学大学院) 藤野清次 (九州大学情報基盤研究開発センター)
Keiichi Murakami (Kyushu Univ.) Seiji Fujino (Kyushu Univ.)

1 はじめに

係数行列 $A \in \mathbb{R}^{n \times n}$ が対称な連立 1 次方程式の解法として CG 法が知られている. CG 法を並列化する場合, 各反復で 2 回行われる内積計算が同期点となる. 内積計算では Reduction と呼ばれる演算が行われ, プロセス並列では全体全通信, Thread 並列では逐次的な処理を必要とするため, 並列台数を増やした場合に高速化の妨げとなる.

一方, MrsR (Minimized Residual method based on Shadow three-term Recurrence, ミセス R) 法 [1] では, 各反復で 4 回の内積計算を必要とするが, 内積のために必要な同期は各反復で 1 回で済む. そのため, CG 法よりも優れた並列性を示すと期待できる.

本研究では, CG 法と MrsR 法を並列化し, 両者の収束性と同期回数の違いを明らかにする. 数値実験 (2 種類の計算機上) で, MPI を用いて並列化のときの CG 法と MrsR 法の並列収束性を比較論じることにする.

2 MrsR 法の導出

提案する Minimized Residual method based on Shadow three-term Recurrence (シャドウ三項漸化式に基づく最小残差: MrsR と以下略す) 法では, 残差ベクトルが初期残差ベクトル r_0 と安定化多項式との積で定義される. MrsR 法は CG 法や Bi-CG 法の残差ベクトルが満たす Lanczos 多項式に, 二つのパラメータ ζ_k, η_k を導入して, 次の三項漸化式を満たす多項式 $H_k(\lambda)$ を設計する.

$$H_0(\lambda) = 1, H_1(\lambda) = (1 - \zeta_0 \lambda) H_0(\lambda), \quad (1)$$

$$H_{k+1}(\lambda) = (1 + \eta_k - \zeta_k \lambda) H_k(\lambda) - \eta_k H_{k-1}(\lambda). \quad (k = 1, 2, \dots) \quad (2)$$

MrsR 法の残差ベクトル r_k は, 上記の三項漸化式 (1)-(2) で生成される多項式 $H_k(\lambda)$ を用いて,

$$r_k \equiv H_k(A) r_0 \quad (3)$$

と定義する. 式 (1)-(2) によって生成された多項式 $\{H_k(\lambda)\}$ は, 任意の k に対して $H_k(0) = 1$ を満たすので, $H_{k+1}(0) - H_k(0) = 0$ である. したがって, 次の式を満足する k 次多項式 $\tilde{G}_k(\lambda)$ が存在する.

$$\tilde{G}_{k+1}(\lambda) = \lambda G_k(\lambda) = \frac{H_k(\lambda) - H_{k+1}(\lambda)}{\zeta_k}. \quad (4)$$

このとき、式 (2) は、

$$H_{k+1}(\lambda) - H_k(\lambda) = -\zeta_k \lambda H_k(\lambda) + \eta_k (H_k(\lambda) - H_{k-1}(\lambda)) \quad (5)$$

と書き直せるので、多項式列 $\{\tilde{G}_{k+1}(\lambda)\}$ と多項式列 $\{H_{k+1}(\lambda)\}$ を用いた次の漸化式を生成できる。

$$\tilde{G}_0(\lambda) = 0, \quad H_0(\lambda) = 1, \quad (6)$$

$$\zeta_k \tilde{G}_{k+1}(\lambda) = \zeta_k \lambda H_k(\lambda) + \zeta_{k-1} \eta_k \tilde{G}_k(\lambda), \quad (7)$$

$$H_{k+1}(\lambda) = H_k(\lambda) - \zeta_k \tilde{G}_{k+1}(\lambda). \quad (k = 0, 1, \dots) \quad (8)$$

式 (8) は $H_{k-1}(\lambda)$ の項が隠れた三項漸化式であるので、シャドウ (shadow) 三項漸化式と呼ぶことにする。補助ベクトル \mathbf{y}_k

$$\mathbf{y}_k \equiv \zeta_{k-1} A G_{k-1}(A) \mathbf{r}_0 = \zeta_{k-1} \tilde{G}_k(A) \mathbf{r}_0 \quad (9)$$

と、式 (6)-(8) から、残差ベクトル $\mathbf{r}_{k+1} = H_{k+1}(A) \mathbf{r}_0$ を計算する漸化式が次のように得られる。

$$\mathbf{y}_{k+1} = \eta_k \mathbf{y}_k + \zeta_k A \mathbf{r}_k, \quad (10)$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{y}_{k+1}. \quad (11)$$

また、パラメータ ζ_k, η_k は次の式 (12) から各々導出する。

$$\|\mathbf{r}_{k+1}\|_2 = \min_{\zeta_k, \eta_k} \|\mathbf{r}_k - \eta_k \mathbf{y}_k - \zeta_k A \mathbf{r}_k\|_2. \quad (12)$$

式 (12) が、最小となるようなパラメータ ζ_k は、

$$\zeta_k = \frac{(\mathbf{r}_k, A \mathbf{r}_k) - (\mathbf{y}_k, A \mathbf{r}_k) \eta_k}{(A \mathbf{r}_k, A \mathbf{r}_k)} \quad (13)$$

となる。同様にして、パラメータ η_k は、

$$\eta_k = \frac{(\mathbf{r}_k, \mathbf{y}_k) - (\mathbf{y}_k, A \mathbf{r}_k) \zeta_k}{(\mathbf{y}_k, \mathbf{y}_k)} \quad (14)$$

となる。

次に、式 (13) に式 (14) を代入して、パラメータ ζ_k について整理すると、

$$\zeta_k = \frac{(\mathbf{y}_k, \mathbf{y}_k)(\mathbf{r}_k, A \mathbf{r}_k) - (\mathbf{y}_k, A \mathbf{r}_k)(\mathbf{r}_k, \mathbf{y}_k)}{(\mathbf{y}_k, \mathbf{y}_k)(A \mathbf{r}_k, A \mathbf{r}_k) - (\mathbf{y}_k, A \mathbf{r}_k)(A \mathbf{r}_k, \mathbf{y}_k)} \quad (15)$$

が得られる (ただし、分母は 0 でないとする)。

同様に、パラメータ η_k についても、式 (14) に式 (13) を代入して整理すると、

$$\eta_k = \frac{(A \mathbf{r}_k, A \mathbf{r}_k)(\mathbf{y}_k, \mathbf{r}_k) - (\mathbf{y}_k, A \mathbf{r}_k)(A \mathbf{r}_k, \mathbf{r}_k)}{(\mathbf{y}_k, \mathbf{y}_k)(A \mathbf{r}_k, A \mathbf{r}_k) - (\mathbf{y}_k, A \mathbf{r}_k)(A \mathbf{r}_k, \mathbf{y}_k)} \quad (16)$$

が得られる.

また, ベクトル $\mathbf{r}_k, \mathbf{y}_k$ は, 式 (12) から次の性質を満たす.

$$(\mathbf{r}_{k+1}, A\mathbf{r}_k) = 0, \quad (17)$$

$$(\mathbf{r}_{k+1}, \mathbf{y}_k) = 0. \quad (18)$$

式 (18) に, 式 (11) を代入すると, 次の等式が得られる.

$$(\mathbf{r}_k, \mathbf{y}_k) = (\mathbf{y}_{k+1}, \mathbf{y}_k). \quad (19)$$

内積 $(\mathbf{r}_{k+1}, \mathbf{y}_{k+1})$ に, 式 (10) を代入し, さらに, 式 (17), (18) を用いると,

$$\begin{aligned} (\mathbf{r}_{k+1}, \mathbf{y}_{k+1}) &= (\mathbf{r}_{k+1}, \eta_k \mathbf{y}_k + \zeta_k A\mathbf{r}_k) \\ &= \eta_k (\mathbf{r}_{k+1}, \mathbf{y}_k) + \zeta_k (\mathbf{r}_{k+1}, A\mathbf{r}_k) = 0 \end{aligned} \quad (20)$$

となるので, ベクトル $\mathbf{r}_k, \mathbf{y}_k$ は, 次の性質を満たす.

$$(\mathbf{r}_k, \mathbf{y}_k) = (\mathbf{y}_{k+1}, \mathbf{y}_k) = 0. \quad (21)$$

さらに, 内積計算 $(\mathbf{y}_k, \mathbf{y}_k)$ は次のように変形できる.

$$\begin{aligned} (\mathbf{y}_k, \mathbf{y}_k) &= (\mathbf{r}_{k-1} - \mathbf{r}_k, \eta_{k-1} \mathbf{y}_{k-1} + \zeta_{k-1} A\mathbf{r}_{k-1}) \\ &= \eta_{k-1} (\mathbf{r}_{k-1}, \mathbf{y}_{k-1}) + \zeta_{k-1} (\mathbf{r}_{k-1}, A\mathbf{r}_{k-1}) - \eta_{k-1} (\mathbf{r}_k, \mathbf{y}_{k-1}) - \zeta_{k-1} (\mathbf{r}_k, A\mathbf{r}_{k-1}) \\ &= \zeta_{k-1} (\mathbf{r}_{k-1}, A\mathbf{r}_{k-1}). \end{aligned} \quad (22)$$

最後に, 近似解を構成するために次の補助ベクトル \mathbf{p}_k を導入する.

$$\mathbf{p}_k \equiv A^{-1} \tilde{G}_{k+1}(A) \mathbf{r}_0. \quad (23)$$

このとき, 式 (6)-(8) は次のように表現できる.

$$\mathbf{p}_k = \mathbf{r}_k + \frac{\zeta_{k-1}}{\zeta_k} \eta_k \mathbf{p}_{k-1}, \quad (24)$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \zeta_k A\mathbf{p}_k. \quad (25)$$

また, $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ の関係より, 近似解 \mathbf{x}_{k+1} は次のように求めることができる.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \zeta_k \mathbf{p}_k. \quad (26)$$

3 数値実験

3.1 計算機環境と計算条件

Dell PowerEdge と PRIMERGY の 2 種類の計算機上で数値実験を行った. 表 1 に 2 種類の計算機環境と表 2 テスト行列の主な特徴を示す. 表中の “nnz” は行列の総非零要素数, “ave. nnz” は 1 行当りの平均非零要素数を各々表す.

表 1: 2 種類の計算機環境

computer	Dell PowerEdge	PRIMERGY
OS	Scientific Linux	Red Hat Linux
CPU	Intel Xeon E3-1220	Intel Xeon X5670
clock	3.10GHz	2.93GHz
memory	8GB	48GB
compiler	Inter Fortran ver.12.0.4	Fujitsu Parallelnavi (Fortran)
cores/node	4	12
network	Gigabit Ethernet	InfiniBand

表 2: テスト行列 (10 個) の主な特徴

matrix	n	nnz	ave. nnz	analytic field
s2	19,800	67,036	3.4	structural
ccs20	438,440	3,733,629	8.5	electromag.
oilpan	73,752	2,148,558	29.1	structural
s3dkt3m2	90,449	3,686,223	40.8	
s3dkq4m2	90,449	4,427,725	49.0	
vanbody	47,072	2,329,056	49.5	structural
BenElechi1	245,874	13,150,496	53.5	
olafu	16,146	1,015,156	62.9	
bmwcra_1	148,770	10,641,602	71.5	
x104	108,384	8,713,602	80.4	

計算条件は以下の通りである。プログラムは Fortran90 により実装し、並列化は MPI を用いて行った。コンパイルオプションは Dell PowerEdge では “-fast”, PRIMERGY では “-Kfast” とした。計算は全て倍精度浮動小数点演算で行った。収束判定値は相対残差の 2 ノルム： $\|\mathbf{r}_{k+1}\|_2/\|\mathbf{r}_0\|_2 \leq 10^{-8}$ とした。初期近似解 \mathbf{x}_0 はすべて 0, 最大反復回数は 10 万回とした。行列は予め対角スケーリングによって対角項を 1 に正規化した。右辺項は物理条件から得られる値とした。時間計測は C 言語の関数 “gettimeofday” を用いて行い、計測結果は 5 回の計測のうち最大値, 最小値を除く 3 回の平均とした。

3.2 実験結果

表 3 に Dell PowerEdge における MPI により並列化した CG 法と MrsR 法の収束性を示す。表中の “np” はプロセス数, “Av” は行列 A とベクトル \mathbf{v} の積の計算時間, “その他”

は行列ベクトル積計算以外が時間の全体に占める割合 [単位 : %] を各々意味する。

表 3: Dell PowerEdge における MPI により並列化した CG 法と MrsR 法の収束性

行列	解法	np	反復 回数	時間 [sec.]			ratio	台数 効果	通信比 率 [%]
				<i>Av</i>	その他	合計			
s3dkt3m2	CG	1	40,602	154.94	11.56	166.50	-	1.00	6.94
		4	40,595	44.91	8.09	53.00	-	3.14	15.27
		8	40,591	27.40	7.43	34.83	-	4.78	21.33
		16	40,589	19.10	8.99	28.09	-	5.93	32.01
	MrsR	1	19,845	76.37	7.22	83.59	0.50	1.00	8.64
		4	19,833	22.89	2.64	25.53	0.48	3.27	10.34
		8	19,828	13.04	1.72	14.76	0.42	5.66	11.64
		16	19,828	9.02	2.01	11.03	0.39	7.58	18.23
s3dkq4m2	CG	1	28,724	136.43	8.23	144.66	-	1.00	5.69
		4	28,727	38.37	6.00	44.37	-	3.26	13.52
		8	28,725	22.04	5.42	27.46	-	5.27	19.73
		16	28,723	13.90	6.60	20.51	-	7.05	32.20
	MrsR	1	14,846	70.91	5.41	76.32	0.53	1.00	7.09
		4	14,842	20.59	2.15	22.74	0.51	3.36	9.46
		8	14,840	11.81	1.40	13.21	0.48	5.78	10.60
		16	14,839	7.27	1.55	8.81	0.43	8.66	17.53

表 4-表 5 に Dell PowerEdge における CG 法と MrsR 法の計算時間比較を示す。表中の “ratio” は CG 法の計算時間を 1 としたときの MrsR 法の計算時間の比を表す。表 6 に PRIMERGY での MPI により並列化した CG 法と MrsR 法の収束性を示す。下線を付けた数字は比が 1.0 よりも大きいことを表す。

表 6 より、PRIMERGY では Dell PowerEdge 上での実験結果に比べ、全体的に行列ベクトル積以外に要した時間が短い。台数効果の値は 32 プロセスまで CG 法と MrsR 法で同程度であり、プロセス数が 64、及び 96 の場合は MrsR 法の方が大きいことがわかった。表 7-に表 8 に PRIMERGY における CG 法と MrsR 法の計算時間比較を示す。

図 1 に PRIMERGY 上での CG 法と MrsR 法の台数効果を示す。

4 まとめ

MPI による並列化を行った CG 法と同 MrsR 法の並列収束性を比較した。その結果、MrsR 法が CG 法よりも収束性がよく、プロセス数を多くした場合はさらに並列性能の差が開くことがわかった。

参考文献

- [1] S. Fujino, K. Murakami: A proposal of MrsR method with one global synchronization per one iteration, PMAA'12, Birkbeck Univ. of London, 28-30 June, 2012.

表 4: Dell PowerEdge における CG 法と MrsR 法の計算時間比較 (1 プロセスの場合)

行列	time[sec.]		ratio
	CG	MrsR	
s2	1.82	1.99	<u>1.10</u>
ccs20	20.58	17.56	0.85
oilpan	71.42	31.67	0.44
s3dkt3m2	166.50	83.59	0.50
s3dkq4m2	144.66	76.32	0.53
vanbody	38.21	20.01	0.52
BenElechi1	432.79	136.83	0.32
olafu	27.39	24.10	0.88
bmwcra_1	59.70	64.39	<u>1.08</u>
x104	437.15	329.55	0.75

表 5: Dell PowerEdge における CG 法と MrsR 法の計算時間比較 (16 プロセスの場合)

行列	time[sec.]		ratio
	CG	MrsR	
s2	0.57	0.43	0.76
ccs20	4.53	3.59	0.79
oilpan	12.38	4.42	0.36
s3dkt3m2	28.09	11.03	0.39
s3dkq4m2	20.51	8.81	0.43
vanbody	10.24	4.35	0.42
BenElechi1	44.42	12.59	0.28
olafu	12.30	8.15	0.66
bmwcra_1	15.03	14.59	0.97
x104	61.94	41.52	0.67

表 6: PRIMERGY における MPI により並列化した CG 法と MrsR 法の収束性

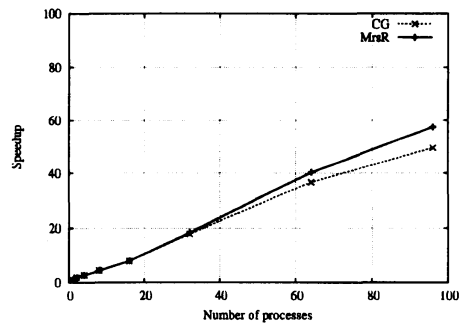
行列	解法	np	反復 回数	時間 [sec.]			ratio	台数 効果	通信比 率 [%]
				Av	その他	合計			
s3dkt3m2	CG	1	40,593	348.04	17.74	365.77	-	1.00	4.85
		16	40,594	41.43	2.84	44.27	-	8.26	6.42
		32	40,588	18.14	1.75	19.88	-	18.40	8.78
		96	40,596	5.27	3.20	8.47	-	43.19	37.73
	MrsR	1	19,836	170.07	10.37	180.45	0.49	1.00	5.75
		16	19,824	21.00	1.14	22.14	0.50	8.15	5.16
		32	19,825	9.00	0.51	9.51	0.48	18.98	5.39
		96	19,825	2.58	0.91	3.49	0.41	51.75	25.98
s3dkq4m2	CG	1	28,724	309.69	12.54	322.23	-	1.00	3.89
		16	28,727	38.57	2.17	40.75	-	7.91	5.33
		32	28,722	16.67	1.40	18.06	-	17.84	7.72
		96	28,726	4.33	2.18	6.51	-	49.50	33.51
	MrsR	1	14,844	160.09	7.76	167.85	0.52	1.00	4.62
		16	14,838	20.22	0.94	21.16	0.52	7.93	4.46
		32	14,838	8.77	0.43	9.20	0.51	18.24	4.72
		96	14,838	2.27	0.66	2.92	0.45	57.40	22.54

表 7: PRIMERGY における CG 法と MrsR 法の計算時間比較 (1 プロセスの場合)

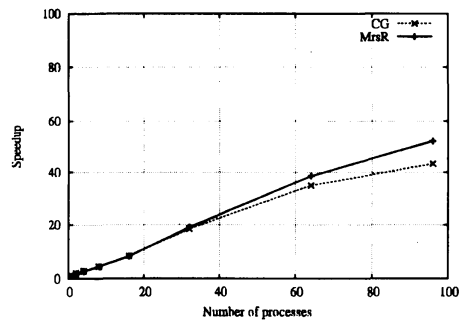
行列	time[sec.]		ratio
	CG	MrsR	
s2	3.98	4.29	<u>1.08</u>
ccs20	38.90	32.64	0.84
oilpan	155.22	67.89	0.44
s3dkt3m2	365.77	180.45	0.49
s3dkq4m2	322.23	167.85	0.52
vanbody	82.04	42.32	0.52
BenElechi1	953.36	298.24	0.31
olafu	58.78	51.02	0.87
bmwcra_1	130.40	134.02	<u>1.03</u>
x104	957.23	716.77	0.75

表 8: PRIMERGY における CG 法と MrsR 法の計算時間比較 (96 プロセスの場合)

行列	time[sec.]		ratio
	CG	MrsR	
s2	0.23	0.21	0.89
ccs20	1.48	1.15	0.78
oilpan	3.62	1.42	0.39
s3dkt3m2	8.47	3.49	0.41
s3dkq4m2	6.51	2.92	0.45
vanbody	2.85	1.28	0.45
BenElechi1	16.26	5.02	0.31
olafu	3.58	2.45	0.68
bmwcra_1	2.97	2.80	0.94
x104	16.49	11.51	0.70



(a) 行列 s3dkq4m2



(b) 行列 s3dkt3m2

図 1: PRIMERGY 上での CG 法と MrsR 法の台数効果